

Estudio de los métodos de estimación: AHP y redes Bayesianas

Joseba Esteban López , José Javier Dolado

Departamento de Lenguajes y Sistemas

Universidad del País Vasco U.P.V./E.H.U.

jesteban004@ikasle.ehu.es , dolado@si.ehu.es

Resumen

Existen multitud de métodos de estimación para el cálculo del esfuerzo en el desarrollo del software. No obstante, los resultados de estos métodos no resultan satisfactorios o son demasiado dependientes del dominio del problema. En este artículo se exponen dos métodos que incluyen el conocimiento experto como base en sus estimaciones: AHP y redes Bayesianas.

1. Introducción

La estimación y medición desempeña un papel muy importante dentro de la creación y mantenimiento de productos y sistemas software. Las estimaciones y las medidas se suelen utilizar para evaluar la viabilidad de un proyecto, analizar sus alternativas, predecir el desarrollo y determinar la cantidad de recursos necesarios. Una buena estimación es clave, y se encuentra acotada por los costes de tiempo y dinero.

El aprovechar la experiencia anterior de proyectos software similares, contribuye a mejorar la exactitud de las estimaciones. Estas últimas suelen requerirse en las etapas iniciales del proyecto software. En estas fases iniciales los proyectos no suelen estar totalmente definidos y se van refinando según se va desarrollando el proyecto. La falta de disponibilidad de esta información previa incrementa la dificultad de conseguir buenas estimaciones, lo que a su vez provoca que los estimadores pasen por alto algunos componentes o actividades del proyecto, además de tener

en cuenta supuestos inválidos. Es por esto que sería interesante poder realizar estimaciones en las fases iniciales del proyecto software, sin disponer de un gran número de datos previos y sin que se vea afectada la exactitud de las estimaciones.

Uno de los retos a los que se deben enfrentar los gestores, se produce a raíz de la aparición de cambios rápidos e impredecibles que afectan a sus estimaciones. Los cambios ocurren inevitablemente a lo largo del ciclo de vida del proyecto y en cualquier área del mismo. Por tanto, el equipo del proyecto se ve obligado a realizar continuas mediciones durante el ciclo de vida del proyecto. Comparar estas mediciones con los valores del objetivo del proyecto permite que el equipo del proyecto pueda lograr el objetivo. Los proyectos software tienen un grado alto de novedad, lo que los caracteriza como difíciles de estimar. La adaptación a los cambios no sería posible sin unas estimaciones lo suficientemente exactas. Unas estimaciones poco exactas o incorrectas afectarían a cualquiera que esté asociado al proyecto, desde los ingenieros y gestores, hasta los clientes, así como a la aceptación del producto, el éxito comercial, el coste operativo y la seguridad.

Hacer estimaciones lo suficientemente exactas conlleva un alto nivel de dificultad. A lo largo de los últimos treinta años, las técnicas de estimación han ido evolucionando, y se han desarrollado diferentes métodos de estimación. Analytic Hierarchy Process es un método de estimación que nos permite realizar estimaciones en fases muy tempranas del proyecto software y sin la necesidad de disponer de datos previos. Las redes bayesianas, en cam-

Definición	Explicación	Valor Realitvo	Valor Recíproco
Mismo tamaño	Las dos entidades tienen aproximadamente el mismo tamaño	1	1.00
Ligeramente mayor (menor)	La experiencia o el juicio reconoce una entidad como algo más grande (más pequeño)	3	0.33
Mayor (menor)	La experiencia o el juicio reconoce una entidad como definitivamente más grande (más pequeño)	5	0.20
Mucho mayor (menor)	El dominio de una entidad sobre otra es evidente; un diferencia muy fuerte de tamaño	7	0.14
Extremadamente mayor (menor)	La diferencia entre las entidades comparadas es de un orden de magnitud	9	0.11
Valores intermedios entre puntos adyacentes de la escala	Cuando el compromiso es necesario	2, 4, 6, 8	0.5, 0.25, 0.16, 0.12

Cuadro 1: Escala verbal propuesta por Thomas L. Saaty

bio, poseen la cualidad de adaptarse a las posibles alteraciones en el proyecto gracias a la realimentación de sus prioridades a posteriori. En este artículo se presentan los dos métodos de estimación: AHP y Redes Bayesianas.

2. Métodos de estimación: AHP y Redes Bayesianas

En esta sección se muestran dos métodos de estimación. Se señalan tanto sus características más importantes como su funcionamiento, así como las características de cada uno de ellos.

2.1. Analytic Hierarchy Process

A pesar de la cantidad de métodos de estimación existentes para prever el tamaño y el coste del software, la mayoría de los gestores de proyectos siguen utilizando los juicios expertos en sus estimaciones. El hecho de que los responsables de la gestión de los proyectos sigan desconfiando de los métodos de estimación puede deberse a diferentes causas, como la falta de información en las fases iniciales de los proyectos software, la especificidad del dominio sobre el que trata el software o el nivel de esfuerzo y tiempo requeridos por determinados métodos de estimación. Por el lado contrario, las estimaciones basadas únicamente en juicios expertos suelen aquejar de falta de exactitud (cómo de cerca está la medida del valor real).

Analytic Hierarchy Process (A.H.P.) es un método de estimación de ayuda a la toma de decisiones basado en múltiples criterios de decisión. AHP fue propuesto por Thomas L. Saaty en la década de los 80. Desde entonces se ha convertido en una de las técnicas más utilizadas para la toma de decisiones multia-tributo. AHP se basa en juicios subjetivos realizados por los expertos. Los expertos aportan su conocimiento subjetivo, consistente en comparaciones entre las principales tareas que constituyen un proyecto software. Los expertos estiman, más que un valor exacto, una medida relativa. Basándose en esta idea, el experto, evaluando la proporción entre cada par de tareas definidas en la aplicación software, consigue una mayor exactitud en sus evaluaciones.

2.1.1. Algoritmo AHP

El algoritmo del método de estimación Analytic Hierarchy Process consta de cinco pasos que exponen a lo largo de este apartado.

En primer lugar se define el problema. Para esto hay que dividirlo en tres partes: objetivo, criterios y alternativas. El objetivo es la decisión que se ha de tomar. Los criterios representan los factores que afectan a la preferencia o deseabilidad de una alternativa. Pueden estar compuestos por otros criterios o subcriterios. Las alternativas son las posibles opciones o acciones de las que se dispone y de las cuales se intenta elegir una. Una alternativa puede ser cualquier entidad relevante en un grupo de interés, como casos de uso, módulos soft-

$$A^{n \times n} = \begin{cases} a_{ij} = \frac{s_i}{s_j} & \text{Cómo de grande o pequeña es la entidad } i \text{ respecto de la entidad } j \\ a_{ij} = 1 & \text{Las entidades } i \text{ y } j \text{ son de la misma proporción} \\ a_{ij} = \frac{1}{a_{ji}} & \text{Inversamente proporcionales} \end{cases} \quad (1)$$

ware, objetos etc., es decir, cualquier entidad de la que se pueda conocer las magnitudes que se necesitan a la hora de tomar una decisión. Una vez hecho esto, se debe construir la jerarquía, de la que AHP toma el nombre. Se han de tomar una serie de decisiones identificando qué factores son importantes y cómo interactúan con el resto de factores.

Aunque no se trate de una parte esencial de la metodología de AHP, establecer una escala verbal o *verbal scale* agiliza el proceso de estimación y no hace peligrar la exactitud de la estimación. La escala verbal ayuda a entender cómo de menor es el término "menor que" ó cómo de mayor es el término "mayor que". Se compone de cuatro atributos: definición o etiqueta, explicación, valor relativo y valor recíproco. La escala verbal establece un consenso que evita que los expertos o los participantes en la estimación pierdan tiempo discutiendo sobre el grado de diferencia entre las alternativas comparadas. Thomas L. Saaty [1] nos propone una escala compuesta por 9 valores y sus recíprocos, como se puede ver en el Cuadro 1.

Con la jerarquía y la escala verbal ya bien definidas, se pasa a obtener la matriz de juicios o *judgement matrix*. Es en esta etapa donde entra en juego el juicio experto. El experto, basándose en la escala verbal, debe hacer comparaciones por parejas en cada nivel de la jerarquía, e ir anotándolos en la matriz. La matriz de juicios es de tamaño $n \times n$, siendo n el número de alternativas de las que se dispone. Cada celda de la matriz de juicios contiene un valor a_{ij} , que representa el tamaño relativo de la entidad i respecto del de la entidad j . Los elementos de la matriz se definen como se muestra en (1). Si la entidad i es a_{ij} veces mayor (o menor) que la entidad j , entonces la entidad j es $1/a_{ij}$ veces menor (o mayor) que la entidad i . Teniendo en cuenta esta premisa y

que la diagonal de la matriz sólo tiene como valor la unidad, no haría falta calcular todos los valores de la matriz. Sólo es necesario calcular una mitad de la matriz, ya sea la parte superior a la diagonal o la inferior.

A la hora de hacer las comparaciones por parejas, es necesaria la colaboración del experto y al menos una entidad de referencia o *reference task* de la que se conozca su magnitud real de proyectos anteriores. Las proporciones de la entidad de referencia son las primeras que se han de situar en la matriz. Es importante que la proporción de esta entidad no ocupe los valores extremos de la escala verbal, sino que se sitúe más o menos hacia la mitad de la escala. De esta manera se minimizan los posibles prejuicios introducidos en la matriz de juicios. Otra posibilidad, con el mismo objetivo, radica en introducir más de una entidad de referencia repartidas uniformemente en la escala verbal.

En este punto, con una matriz de juicios por cada criterio a tener en cuenta en la toma de decisión, se calcula la escala de proporción o *ratio scale*. La escala de proporción es un vector r en el que cada posición del vector contiene un valor proporcional a la entidad i en relación al criterio elegido. También se calcula el índice de inconsistencia o *inconsistency index*. Este índice nos proporciona una medida de cómo de lejos está nuestra estimación de la consistencia perfecta. Una matriz de juicios perfectamente consistente es aquella en la que todos sus elementos satisfacen $a_{ij} \times a_{jk} = a_{ik} \forall i, j, k$. Como procedimiento para calcular la escala de proporción y el índice de inconsistencia, proponemos la utilización del modelo propuesto por Eduardo Miranda en [1], por sencillez y buenos resultados. Hay que calcular la media geométrica v_i de cada fila de la matriz de juicios definida para un determinado criterio. El valor de v_i viene dado por (2). La escala de proporción,

denominada vector de valores propios o *eigenvalue*, r , consiste en un vector en el que cada valor se calcula aplicando (3).

$$v_i = \sqrt[n]{\prod_{j=1}^n a_{ij}} \quad (2)$$

$$r = [r_1, r_2, \dots, r_n] \text{ con } r_i = \frac{v_i}{\sum_{j=1}^n v_j} \quad (3)$$

El índice de inconsistencia se puede calcular de la siguiente forma (4).

$$CI = \frac{\sqrt{\sum_{i=1}^n \sum_{j>i}^n \left(\ln a_{ij} - \ln \frac{v_i}{v_j} \right)^2}}{\frac{(n-1) \times (n-2)}{2}} \quad (4)$$

A partir del vector de valores propios o *eigenvalue* y el valor real de la entidad de referencia, se puede calcular el valor absoluto de cada entidad. Para ello se debe aplicar la expresión (5). Aplicando estos cálculos con el resto de criterios, previamente ordenados según el porcentaje de contribución sobre el proyecto en su totalidad, se puede calcular el valor de cada alternativa.

$$Valor_i = \frac{r_i}{r_{referencia}} \times Valor_{referencia} \quad (5)$$

En el siguiente apartado, se muestran las características más relevantes de AHP. También se indican los principales problemas que presenta este método de estimación.

2.1.2. Características de AHP

Analytic Hierarchy Process es uno de los métodos de estimación más sencillos cuya mayor dificultad radica en identificar los atributos y su contribución relativa. Además, proporciona una visión del proyecto software jerarquizada, estructurada y sistemática. Asimismo, AHP es poco propenso a errores, permitiendo estimaciones precisas con hasta un 40 % de comparaciones erróneas. A pesar de este dato no se puede afirmar que AHP sea mejor que la estimación experta, ya que

está basado, precisamente, en comparaciones hechas por expertos entre pares de tareas.

AHP aporta una notable ventaja para los expertos, ya que resulta más sencillo hacer comparaciones por parejas (entre los pares de tareas) que estimar cada tarea de una en una. Por otro lado, el número de comparaciones que deben realizar puede suponer un problema. Esto se debe a que la relación de las comparaciones a realizar y el número de tareas es de orden cuadrático. Si nuestro proyecto se compusiera de n tareas, el número de comparaciones que se deberían realizar sería de $n(n-1)/2$. En el supuesto de manejar 30 tareas, se deberán realizar 435 comparaciones. Para evitar esto, aún a riesgo de aumentar la homogeneidad de la matriz de juicios, se pueden agrupar las tareas similares en grupos más reducidos.

En las fases iniciales del desarrollo de un proyecto software suele darse una carencia de datos de referencia. El método de estimación AHP resulta muy útil en estas etapas iniciales, ya que como mínimo necesita un único dato de referencia: la tarea de referencia. Esta característica nos posibilita hacer estimaciones bastante precisas en fases tempranas del desarrollo de un proyecto software. Es importante que el porcentaje de contribución de la tarea de referencia al proyecto global sea lo más preciso posible. Esto aumentará la exactitud de las estimaciones del resto de las tareas. En caso contrario, podrá derivar en errores. La exactitud de las predicciones también se verá mejorada con el uso de más de una tarea de referencia.

Con el método de estimación AHP puede darse el fenómeno del Rank Reversal o alteración del rango. Este fenómeno consiste en un cambio en el ranking relativo de las tareas, al introducir una nueva tarea o eliminar una de ellas. Los autores Ying-Ming Wang y Taha M.S. Elhag en [2], exponen la existencia de varias propuestas enfocadas a evitar el fenómeno de la alteración en el rango. Estas propuestas tienen en cuenta si la alternativa introducida o eliminada del conjunto de alternativas seleccionadas agrega o no información. En ese mismo artículo se expone la propuesta de los autores, en la que se preserva el ran-

king de las alternativas sin necesidad de variar los pesos de las alternativas ni el número de criterios. Para esto, los autores proponen la normalización del vector de valores propios o *eigenvalue*.

Un aspecto a tener en cuenta utilizando el método de estimación AHP radica en la escala elegida, tanto por su proporción como por el número de puntos que la constituyen. En AHP el éxito en las estimaciones reside en la exactitud de las comparaciones entre las tareas. Estas comparaciones, son consecuencia directa de la escala de evaluación elegida, así como del número de puntos de la escala. La escala propuesta por Thomas L. Saaty en [1], propone el uso de una escala que varía entre 1/9 y 9, lo que supone un total de 17 puntos en la escala. En [3] los autores aconsejan el uso de una escala con un número de etiquetas más bajo, ya que esto nos facilitará el manejo de la escala. Eduardo Miranda en [1] también propone una escala diferente. Ésta consta de menos puntos que la propuesta por Saaty, y según el autor, es más fácil de utilizar por los expertos.

2.2. Redes Bayesianas

En este estudio se incluyen las redes Bayesianas dada su cada vez mayor popularidad dentro de la ingeniería del software, habiendo sido utilizadas en diferentes áreas como la estimación del esfuerzo y calidad, pruebas de software, fiabilidad e interfaces gráficas. Las redes Bayesianas son modelos gráficos probabilísticos que representan una función de distribución conjunta sobre un conjunto finito de variables. Formalmente, una red Bayesiana es un grafo dirigido acíclico.

El gráfico está dividido en una parte cualitativa y otra cuantitativa. La parte cualitativa, es el grafo en sí. El grafo está compuesto de nodos y arcos entre los nodos. Los nodos representan variables aleatorias del dominio X_1, X_2, \dots, X_n , mientras que los arcos representan valores de dependencias entre las variables. Las redes Bayesianas asumen que un nodo depende únicamente de sus padres.

La parte cuantitativa representa la incertidumbre del problema por medio de probabilidades condicionadas: posibles relaciones causa

efecto entre los nodos. Cada nodo posee una tabla de probabilidades condicionales asociada, que define la probabilidad de cada uno de los estados en los que puede estar una variable, dados los posibles estados de sus padres. Esta tabla de probabilidades representa los posibles valores (estados) que puede tomar esa variable y la probabilidad de que se dé cada uno de ellos. Una red Bayesiana muestra la probabilidad de distribución conjunta para un grupo de variables X_1, X_2, \dots, X_n (6), tal que el valor que toma la variable X_i viene representado por x_i . Los valores que tienen el conjunto de los padres de la red Bayesiana del nodo X_i se representa con $padres(X_i)$. Cada estado de una variable se calcula multiplicando un número de valores en las tablas de probabilidades condicionales.

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(x_i | padres(X_i)) \quad (6)$$

Las redes Bayesianas se utilizan en problemas donde no se tiene un completo conocimiento del estado del sistema. Sin embargo, permite realizar observaciones con el fin de actualizar el resto de probabilidades. Cuando se conoce qué valor toma una variable se está hablando de evidencias. Existen dos tipos de evidencias: *firme* o *específica*, y *parcial* o *virtual*. La primera de ellas se da cuando se asigna un valor concreto a una variable. De los posibles valores o estados que puede tomar una variable, cuando uno de ellos tiene un probabilidad del 100%, se denomina *certeza absoluta*. Las evidencias *parciales* o *virtuales* permiten actualizar las probabilidades a priori de los estados que puede tomar una variable, es decir, antes de tener la certeza absoluta de qué valor toma dicha variable.

Las variables continuas en las redes Bayesianas deben ser adaptadas al hecho de que cada variable sólo dispone de un conjunto de estados finitos en las tablas de probabilidad condicional. Una manera de abordar este problema radica en discretizar las variables continuas. La discretización consiste en dividir el rango de las variables continuas en

intervalos exhaustivos y exclusivos. Esto conlleva una pérdida de información dependiente del dominio y del número de intervalos en los que se divide el rango de la variable continua.

Las probabilidades antes de introducir evidencias en la red Bayesiana, se conocen como *probabilidades a priori*. A partir de las evidencias de que se dispongan, junto con la red Bayesiana bien definida, se pueden calcular o estimar las variables de las que se desconocen sus valores. Una red Bayesiana proporciona un sistema de inferencia, donde una vez encontradas nuevas evidencias sobre el estado de ciertos nodos, se modifican sus tablas de probabilidad condicional y a su vez, las nuevas probabilidades son propagadas al resto de los nodos. Esta propagación de probabilidades recibe el nombre de *inferencia probabilística*. Las nuevas evidencias propagadas se llaman *probabilidades a posteriori*. Ante redes formadas por un gran número de nodos y dependencias, la propagación de probabilidades tiene un coste computacional alto, siendo un problema *NP-complejo*. Actualmente existen algoritmos de propagación de probabilidades con los que es posible modelar redes Bayesianas de problemas reales. Existen dos grandes grupos de algoritmos de propagación: los algoritmos de propagación exactos, cuando no hay error en las probabilidades calculadas, y los algoritmos aproximados, cuando las probabilidades de los nodos son estimadas con cierto margen de error.

2.2.1. Creación de redes Bayesianas

Para crear una red Bayesiana que modele un problema, básicamente hay que definir la estructura (grafo) y los parámetros de la red (tablas de probabilidad). Para esto, hay que seguir los pasos situados bajo estas líneas y realizar varias iteraciones de los mismos.

- Lo primero que se debe hacer, una vez que se tenga pleno conocimiento del dominio, es seleccionar las variables de interés. En dominios complejos puede resultar complicado enumerar todas las variables importantes y conocer sus relaciones causales.

- Una vez seleccionadas las variables de interés, se decide si cada variable del dominio será utilizada como variable de entrada, su tipo y si hay que discretizarla o no.
- El siguiente paso consiste en definir la topología de la red, es decir las relaciones causales entre las variables. Diseñando la topología de la red Bayesiana de esta manera se minimiza el número de arcos innecesarios y se maximizan las independencias condicionales.
- Para terminar de crear la red Bayesiana, hay que definir las tablas de probabilidad para cada uno de los nodos de la red. Los ingenieros del conocimiento pueden ayudarse de la minería de datos si existen bases de datos del dominio.
- Una vez creada la red Bayesiana es necesaria su evaluación y verificación de su utilidad. Con este propósito se utiliza el análisis de sensibilidad, que permite comprobar cómo varían los resultados de algunas variables ante cambios en los valores de algunas evidencias de ciertas variables.

A continuación se presenta un ejemplo de red Bayesiana sencilla, con el fin de mostrar el aspecto de las redes Bayesianas. En la (Figura 1) se puede ver la arquitectura de una red Bayesiana para la estimación del esfuerzo software sacado de [4], donde aparece explicado con más detalle. En este caso el nodo denominado esfuerzo software representa la variable dependiente. Este nodo depende de tres nodos padres: metodología de desarrollo, tipo de herramienta CASE y experiencia con las herramientas CASE. En esta figura se muestran los posibles estados en los que se puede encontrar cada variable y su distribución de probabilidad a priori. En el caso del nodo de esfuerzo software se muestra la tabla de probabilidades condicionadas. La distribución de probabilidades a priori se ha obtenido a partir de un conjunto de datos consistente en 33 proyectos software reales de dos grandes compañías del nordeste de Estados Unidos.

Tipo Herramienta	INCASE						IEF						
	Alta		Media		Baja		Alta		Media		Baja		
	NoRAD	RAD	NoRAD	RAD	NoRAD	RAD	NoRAD	RAD	NoRAD	RAD	NoRAD	RAD	
Esfuerzo Software	Alto	0.0	-	0.0	0.0	0.06	-	-	-	0.0	-	1.0	0.0
	Medio	0.5	-	0.0	0.0	0.06	-	-	-	1.0	-	0.0	1.0
	Bajo	0.5	-	1.0	1.0	0.88	-	-	-	0.0	-	0.0	0.0

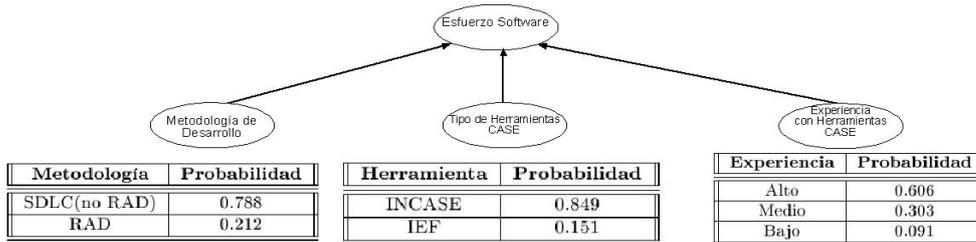


Figura 1: Ejemplo de arquitectura de una red Bayesiana

2.2.2. Características de las redes Bayesianas

La aplicación de las redes Bayesianas en la ingeniería del software ha sido menor que en otras áreas como la medicina. Esto puede deberse a varias causas, como que el tamaño de las bases de datos en ingeniería del software sean más pequeñas, las mediciones sean más subjetivas, las herramientas estén madurando y no sean específicas de la ingeniería del software y, además, que las relaciones causa efecto dentro de los proyectos software sean difíciles de evaluar.

A pesar de esto, las redes Bayesianas poseen una serie de características idóneas para la ingeniería del software. Las redes Bayesianas proveen de una representación gráfica de las relaciones explícitas de dependencia del dominio, de manera que permiten modelar sistemas complejos posibilitando entender las relaciones causales visualizándolas por medio del grafo. Además, están compuestas por dos partes bien diferenciadas, la parte cualitativa y la cuantitativa, que permite utilizar criterios objetivos (otros proyectos finalizados) y subjetivos (la experiencia de los expertos del dominio), respectivamente. Por otro lado, las redes Bayesianas permiten la inferencia bidireccional, con lo se permite hacer inferencia en ambos sentidos. De esta manera se puede predecir que variables de entrada son necesarias para obtener las variables de salida deseadas.

También permiten el uso de valores con grados de certidumbre. Esto permite modelar la incertidumbre de manera efectiva y explícita, por lo que se pueden generar buenas predicciones con información incompleta. Asimismo, la salida que proporciona una red Bayesiana es una probabilidad de distribución en vez de un valor concreto. Este tipo de información es útil a la hora de medir la confianza que se puede depositar en la salida de la red, y más si el modelo va a ser utilizado en la toma de decisiones. No se debe olvidar las facilidades que aportan las redes Bayesianas ante un análisis de sensibilidad, ya que permiten calcular la sensibilidad de ciertas variables de la red, simplemente modificando las evidencias.

Existen una serie de características que suponen una dificultad o limitación a la hora de crear redes Bayesianas. La definición de la estructura de la red Bayesiana puede suponer una tarea compleja incluso para expertos del dominio, pudiendo crear desigualdades entre el modelo construido y el problema del dominio. También se da el problema de que el modelo construido no tenga en cuenta una serie de relaciones poco evidentes que sí estén en el problema del dominio. Además, una mala estimación de los parámetros puede distorsionar toda la red invalidando los resultados. Las redes Bayesianas son sensibles a la inconsistencia, y sólo resultan útiles si se puede confiar en sus resultados.

Normalmente las estimaciones realizadas mediante redes Bayesianas en las fases iniciales del proyecto software raramente son correctas. Esto es debido a que en estas fases tempranas del desarrollo del proyecto software se cuenta con una definición pobre del proyecto, incluso a la hora de decidir que herramientas o metodologías utilizar. Las estimaciones con redes Bayesianas van mejorando durante el ciclo de vida del proyecto software.

3. Conclusiones

En este artículo se ha realizado un estudio de dos métodos de estimación del esfuerzo de desarrollo de software: AHP y redes Bayesianas. En las etapas iniciales de ciclo de vida del proyecto software ambos modelos se ven afectados por la falta de datos históricos. Con AHP se pueden realizar estimaciones precisas con sólo un dato de referencia, mientras que las redes Bayesianas normalmente no son capaces de proporcionar buenas estimaciones en estas fases tempranas del desarrollo del proyecto software. Una vez avanza el proyecto a lo largo del ciclo de vida, las estimaciones proporcionadas mediante redes Bayesianas mejoran notablemente. Ambos métodos se basan en el conocimiento experto: AHP a la hora de realizar las comparaciones y las redes Bayesianas en el diseño de éstas. Ambos métodos han demostrado su eficacia a la hora de estimar diversos tipos de problemas.

Agradecimientos: Este trabajo se ha

desarrollado gracias a la financiación del proyecto TIN2004-06689-C03-01.

Referencias

- [1] Eduardo Miranda. Improving subjective estimates using paired comparisons. *IEEE Software*, 18(1):87–91, Jan/Feb 2001.
- [2] Ying-Ming Wang and Taha M. S. Elhag. An approach to avoiding rank reversal in ahp. *Decis. Support Syst.*, 42(3):1474–1480, 2006.
- [3] Sahrman Barker Martin Shepperd and Martin Aylett. The analytic hierarchy process and data-less prediction. *Empirical Software Engineering Research Group ES-ERG: TR98-04*, 1998.
- [4] Girish H. Subramanian Parag C. Pendharkar and James A. Rodger. A probabilistic model for predicting software development effort. *IEEE Transactions On Software Engineering*, 31(7):615–624, July 2005.
- [5] Daniel Rodriguez and Javier Dolado. *Redes Bayesianas En La Ingeniería del Software*, pages 1–19. To appear in 2007.
- [6] Richard D. Stutzke. *Estimating Software-Intensive Systems: Projects, Products, and Processes*. The SEI Series in Software Engineering. Addison Wesley Professional, <http://sw-estimation.com/index.html>, 1 edition, Apr 26 2005.